What is claimed is:

1   1.   A method for dynamically partitioning a storage system cache among

2   multiple workload classes having different quality-of-service (QoS) requirements,

3   the cache holding data as data pages, the method comprising the steps of:

4   maintaining a history of recently evicted pages for each class;

5   determining a future cache size for the class based on the history and the

6   QoS requirements, the future cache size being different than a current cache size

7   for the class;

8   determining whether the QoS requirements for the class are being met; and

9   adjusting the future cache size to maximize the number of classes in which

10   the QoS requirements are met.

1   2.   The method as recited in claim 1, wherein the step of determining

2   whether the QoS requirements for the class are being met includes the steps of:

3   recording data concerning a QoS requirement for the class; and

4   comparing the recorded data with said QoS requirement.

1   3.   The method as recited in claim 1, wherein the step of determining a

2 future cache size includes the steps of:

3   recording cache hit data in the history of the class;

4   recording the cache size corresponding to the cache hit data; and

5   determining the future cache size based on the cache hit data and the

6 respective cache sizes.

1   4.   The method as recited in claim 1, wherein the step of adjusting the

2 future cache size includes the steps of:

3   increasing the future cache sizes of the classes whose QoS requirements

4 are not met; and

5   decreasing the future cache sizes of the classes whose QoS requirements

6 are met.

1   5.   The method as recited in claim 4, wherein the step of increasing the

2 future cache sizes includes the step of setting a future cache size as a function of

3 current cache size and the difference between cache hit data and corresponding

4 cache sizes.

1　　　　6.　　The method as recited in claim 4, wherein the step of decreasing the

2　future cache sizes includes the step of setting a future cache size as a function of

3　the current cache size, the number of classes and the difference between cache hit

4　data and corresponding cache sizes.

1　　　　7.　　The method as in claim 1 further comprising the step of allocating the

2　cache space to the classes to maximize the overall cache hits if the QoS

3　requirements for all classes are met.

1　　　　8.　　The method as recited in claim 1, wherein the future cache size is

2　adjusted periodically.

1　　　　9.　　The method as recited in claim 1, wherein the future cache size is

2　adjusted continuously on very request for data.

1　　　　10.　　The method as recited in claim 1, wherein the the future cache size is

2　adjusted to maximized the total class objectives.

1   11.   A storage system capable of dynamically partitioning a system cache

2   among multiple workload classes having different quality-of-service (QoS)

3   requirements, the cache holding data as data pages, the system comprising:

4   means for maintaining a history of recently evicted pages for each class;

5   means for determining a future cache size for the class based on the history

6   and the QoS requirements, the future cache size being different than a current

7   cache size for the class;

8   means for determining whether the QoS requirements for the class are being

9   met; and

10   means for adjusting the future cache size to maximize the number of classes

11 in which the QoS requirements are met.

    .

1   12.   The system as recited in claim 11, wherein the means for determining

2   whether the QoS requirements for the class are being met includes:

3   means for recording data concerning a QoS requirement for the class; and

4   means for comparing the recorded data with said QoS requirement.

1   13.   The system as recited in claim 11, wherein the means for determining

2   a future cache size includes:

3   means for recording cache hit data in the history of the class;

4   means for recording the cache size corresponding to the cache hit data; and

5   means for determining the future cache size based on the cache hit data and

6   the respective cache sizes.

1   14.   The system as recited in claim 11, wherein the means for adjusting

2   the future cache size includes:

3   means for increasing the future cache sizes of the classes whose QoS

4   requirements are not met; and

5   means for decreasing the future cache sizes of the classes whose QoS

6   requirements are met.

1   15.   The system as recited in claim 14, wherein the means for increasing

2   the future cache sizes includes means for setting a future cache size as a function

3   of the current cache size and the difference between cache hit data and

4   corresponding cache sizes.

1   16.   The system as recited in claim 14, wherein the means for decreasing

2   the future cache sizes includes means for setting a future cache size as a function

3   of the current cache size, the number of classes, and the difference between cache

4   hit data and corresponding cache sizes.

1   17.   The system as recited in claim 11 further comprising means for

2   allocating the cache space to the classes to maximize the overall cache hits if the

3   QoS requirements for all classes are met.

1   18.   The system as recited in claim 11, wherein the future cache size is

2   adjusted periodically.

1   19.   The system as recited in claim 11, wherein the future cache size is

2   adjusted continuously on very request for data.

1   20.   The system as recited in claim 11, wherein the the future cache size

2   is adjusted to maximized the total class objectives.

1   21.   A computer-program product for use with a storage system for

2   dynamically partitioning a system cache among multiple workload classes having

3   different quality-of-service (QoS) requirements, the cache holding data as data

4   pages, the computer-program product comprising:

5   a computer-readable medium;

6   means, provided on the computer-readable medium, for maintaining a

7   history of recently evicted pages for each class;

8   means, provided on the computer-readable medium, for determining a future

9   cache size for the class based on the history and the QoS requirements, the future

10   cache size being different than a current cache size for the class;

11   means, provided on the computer-readable medium, for determining whether

12   the QoS requirements for the class are being met; and

13   means, provided on the computer-readable medium, for adjusting the future

14   cache size to maximize the number of classes in which the QoS requirements are

15   met.

1   22.   The computer-program product as recited in claim 21, wherein the

2   means for determining whether the QoS requirements for the class are being met

3   includes:

4   means, provided on the computer-readable medium, for recording data

5   concerning a QoS requirement for the class; and

6   means, provided on the computer-readable medium, for comparing the

7   recorded data with said QoS requirement.

1   23.   The computer-program product as recited in claim 21, wherein the

2   means for determining a future cache size includes:

3   means, provided on the computer-readable medium, for recording cache hit

4   data in the history of the class;

5   means, provided on the computer-readable medium, for recording the cache

6   size corresponding to the cache hit data; and

7   means, provided on the computer-readable medium, for determining the

8   future cache size based on the cache hit data and the respective cache sizes.

1   24.   The computer-program product as recited in claim 21, wherein the

2   means for adjusting the future cache size includes:

3   means, provided on the computer-readable medium, for increasing the

4   future cache sizes of the classes whose QoS requirements are not met; and

5   means, provided on the computer-readable medium, for decreasing the

6   future cache sizes of the classes whose QoS requirements are met.

1   25.   The computer-program product as recited in claim 24, wherein the

2   means for increasing the future cache sizes includes means, provided on the

3   computer-readable medium, for setting a future cache size as a function of the

4   current cache size and the difference between cache hit data and corresponding

5   cache sizes.

1   26.   The computer-program product as recited in claim 24, wherein the

2   means for decreasing the future cache sizes includes means, provided on the

3   computer-readable medium, for setting a future cache size as a function of the

4   current cache size, the number of classes, and the difference between cache hit

5   data and corresponding cache sizes.

1   27.   The computer-program product as recited in claim 21 further

2   comprising means, provided on the computer-readable medium, for allocating the

3   cache space to the classes to maximize the overall cache hits if the QoS

4   requirements for all classes are met.

1   28.   The computer-program product as recited in claim 21, wherein the

2   future cache size is adjusted periodically.

1   29.   The computer-program product as recited in claim 21, wherein the

2   future cache size is adjusted continuously on very request for data.

1   30.   The computer-program product as recited in claim 18, wherein the

2   the future cache size is adjusted to maximized the total class objectives.